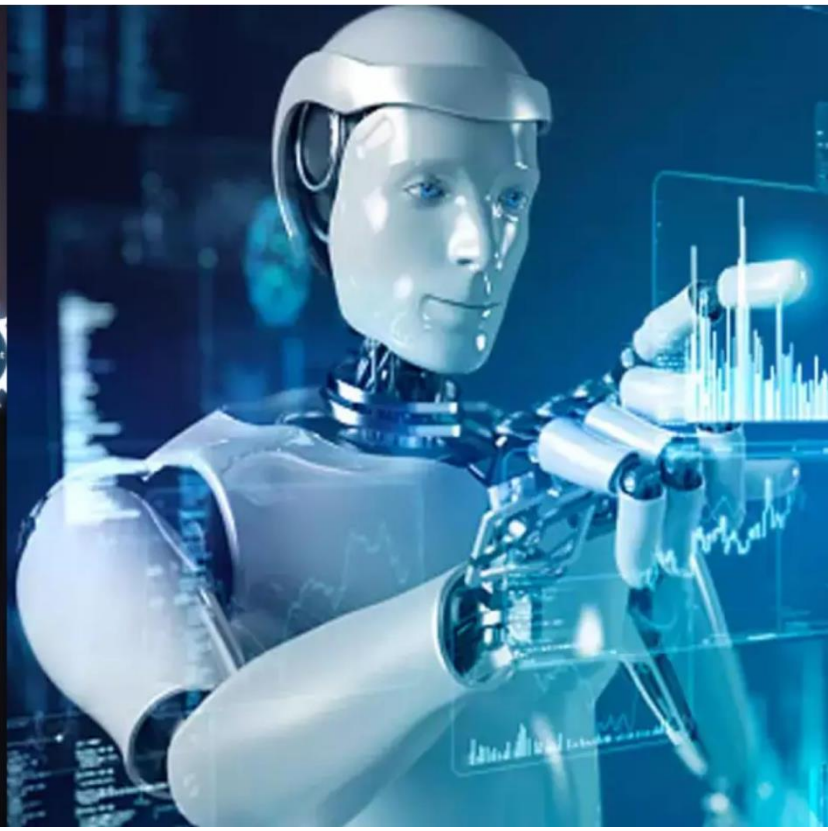
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 4, April 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504044|

Personal Agentic Assistant

B. Avinash [1], A. Sravanthi [2], A. Alekhya [3], G. Sai Charan [4]

Assistant Professor, Department of CSE (AI & ML), ACE Engineering College Hyderabad, India[1].

Department of CSE (AI & ML), ACE Engineering College Hyderabad, India. [2,3,4]

ABSTRACT: This project presents the development of a cross-platform smart personal assistant platform designed to enhance productivity for working professionals using a multi-agent AI-based architecture. The system is built with Flutter for the frontend to support Android, iOS, Web, and desktop platforms, while Fast API is used for building a scalable and high-performance backend. Lang Graph is integrated to enable dynamic orchestration of multiple intelligent agents, and MongoDB is used for flexible and efficient data storage. A central Manager Agent processes user voice or text inputs, analyzes intent, and decomposes complex requests into structured workflows. These workflows are executed by specialized sub-agents responsible for tasks such as document handling, data analysis, software development, communication, and scheduling, through integration with various cloud-based tools and APIs. The platform features an interactive AI avatar, real-time task progress updates, secure authentication, and encrypted communication. The proposed system provides an intelligent, modular, and scalable assistant capable of automating complex professional workflows and improving overall efficiency.

KEYWORDS: Personal Agentic Assistant , Multi-Agent System , Large Language Models , Workflow Automation , Autonomous AI , Cross-Platforms , LangGraph , Flutter , FastAPI **.**

I. INTRODUCTION

Personal Agentic Assistant is an advanced AI-powered **multi-agent smart assistant platform** designed for modern working professionals. It goes beyond traditional chatbots by intelligently understanding user goals, breaking them into tasks, and autonomously executing them using specialized AI agents.

Powered by **Lang Graph**, a central **Manager Agent** dynamically coordinates multiple domain-specific sub-agents (Developer, Data Analyst, Manager, Designer, etc.) to complete complex workflows across tools like **Google Workspace, Microsoft 365, GitHub, Slack, and Cloud APIs**.

Built with **Flutter (cross-platform UI)**, **Fast API (scalable backend)**, and **MongoDB (secure data storage)**, the system delivers a **real-time, voice-enabled AI avatar** with a floating assistant experience across **Web, Mobile, and Desktop**—acting as a personal AI operating system for productivity.

II. LITERATURE SURVEY

[1] Title: *Personal Agentic Assistants: A Survey on Autonomous AI Systems*
Authors: R. Sutton, J. Kober, T. Lillicrap (IEEE Intelligent Systems, 2023)
This study defined that conventional virtual assistants rely heavily on predefined rules and limited intent recognition, making them unsuitable for complex, long-term autonomous tasks. They struggle with reasoning, adaptation, and coordination across multiple tools and environments. The authors introduced an agent-based framework combining reinforcement learning, large language models, and memory-driven reasoning.

[2] Title: Large Language Models as Autonomous Agents
Authors: S. Park, Y. Kim, J. Lee (ACM Computing Surveys, 2024)This research highlighted that purely LLM-powered assistants lack structured planning and often fail to reliably execute long-horizon tasks due to hallucinations and limited control mechanisms. The authors proposed an agentic architecture where LLMs are augmented with external planners, tool APIs, and memory modules to enable strict goal-oriented behavior.

[3] Title: Multi-Agent Systems for Intelligent Personal Assistants
Authors: M. Wooldridge, N. Jennings (Springer – Autonomous Agents and Multi-Agent Systems, 2022)
The authors identified that single-agent assistants face severe scalability issues when handling diverse and concurrent user tasks across different domains. They introduced a multi-agent collaboration model where specialized agents handle distinct tasks such as scheduling, communication, information retrieval, and decision-making.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504044|

OBJECTIVES

The primary objective of this project is to develop a highly intelligent, cross-platform personal assistant capable of automating complex professional workflows. To achieve this overarching goal, the project targets the following specific objectives:

- **To Develop a Cross-Platform Interface:** Build a responsive user interface using Flutter that operates seamlessly across Web, Android, iOS, and Desktop environments, allowing users to seamlessly input voice and text commands.
- **To Implement Multi-Agent Orchestration:** Utilize LangGraph to create a central "Manager Agent" that accurately analyzes user intent and decomposes complex, multi-step goals into structured, executable micro-tasks.
- **To Deploy Specialized Domain Sub-Agents:** Create dedicated sub-agents (such as Developer, Data Analyst, Manager, and Designer agents) to handle specific tasks autonomously based on the Manager Agent's routing and task breakdown.
- **To Integrate External Cloud Tools:** Ensure seamless API integration with third-party professional tools such as Google Workspace, Microsoft 365, GitHub, and Slack to execute real-world workflows.
- **To Ensure Secure and Scalable Operations:** Utilize FastAPI and MongoDB to build a high-performance backend that guarantees secure data handling, fast asynchronous response times, and real-time task monitoring

.

III. METHODOLOGY

Manager Agent and Sub-Agents (LangGraph):

The core intelligence of the system relies on a multi-agent architecture orchestrated by **LangGraph**. Unlike standard linear LLM scripts, LangGraph allows for cyclical, stateful execution where agents can pass information back and forth.

1. State Management: We defined a central state object that tracks the entire lifecycle of a user request. This state includes the original user input, the current task breakdown, the assigned sub-agent, and the intermediate outputs from external tools.

2. The Manager Agent (The Router): The Manager Agent serves as the entry point for the LLM logic. Its implementation involves:

- **Intent Analysis:** Using a prompt structure that forces the LLM to classify the user's goal (e.g., "Schedule a meeting" vs. "Analyze this code").
- **Task Decomposition:** Breaking complex goals into sequential or parallel sub-tasks.
- **Routing:** Directing the specific sub-task to the appropriate specialized node (Sub-Agent) within the LangGraph workflow.

API Integration (Workspace, GitHub, Slack):

For the agents to execute actual workflows, they required access to external platforms. We implemented secure tool-calling functions that the Sub-Agents could invoke.

1. Integration Architecture:

- We used FastAPI to create dedicated endpoints for tool execution.
- The Sub-Agents generate JSON payloads containing the required parameters (e.g., repository name, email address, message body).
- FastAPI receives these payloads, authenticates via stored OAuth tokens or API keys, and makes the secure external HTTP request.

2. Specific Tool Implementation:

- **Google Workspace:** Integrated via the Google Calendar and Gmail APIs. The Scheduling Agent can check availability, create calendar events, and draft summary emails.
- **GitHub:** Integrated via the GitHub REST API. The Developer Agent can fetch repository structures, read specific code files, and create pull requests or issues.
- **Slack:** Integrated via the Slack Web API. The Communication Agent can post updates, send direct messages, and read channel history to provide context.

3. Error Handling: Crucially, the API implementation includes a feedback loop back to the LangGraph orchestrator. If an API call fails (e.g., "Invalid repository name"), the error is caught and sent back to the active Sub-Agent, allowing it to self-correct and attempt the task again.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504044|

Output (Flutter UI & AI Avatar):
The frontend was developed using **Flutter** to ensure a single codebase could deploy natively to Android, iOS, Web, and Desktop environments.

1. User Interaction Flow:

- The interface provides text input fields and a microphone button for voice interaction.
- We utilized standard speech-to-text libraries to capture voice input and convert it into a text string before sending it to the FastAPI backend.

2. Real-Time Feedback Dashboard: Because multi-agent workflows (like analyzing code and sending an email) can take several seconds, the UI was implemented to provide real-time status updates. As the Manager Agent routes tasks and Sub-Agents execute API calls, the backend streams status updates via WebSockets back to the Flutter app.

3. Interactive AI Avatar: To enhance user engagement, we integrated a floating AI Avatar experience. The avatar is visually reactive:

- It displays a "Listening" animation when capturing voice.
- It displays a "Thinking/Processing" animation while the LangGraph orchestrator is running.
- It displays a "Speaking" animation while reading the final compiled response back to the user via Text-to-Speech (TTS).

4. Data Persistence (MongoDB): All user sessions, task histories, and execution logs are saved to a MongoDB instance. This allows the user to review past actions, view the specific steps the agents took, and download final results (like generated reports or code snippets) directly from the UI.

IV. PROPOSED SYSTEM

The proposed system leverages a multi-agent AI architecture powered by LangGraph, functioning as an advanced smart personal assistant platform designed specifically to enhance the productivity of modern working professionals. Instead of relying on a single, static Large Language Model (LLM) to merely answer queries like a traditional chatbot, the proposed system uses a central Manager Agent to intelligently understand overarching user goals, break them down into actionable steps, and autonomously execute them using specialized sub-agents.

System Overview
The proposed system include The **Personal Agentic Assistant** is an advanced, AI-driven digital platform designed to automate complex, multi-step workflows for modern working professionals. Moving beyond the capabilities of traditional reactive chatbots, this system acts as a proactive, autonomous AI operating system. It leverages a multi-agent architecture to understand high-level user goals, decompose them into manageable micro-tasks, and execute them across various enterprise applications.

The system is built upon a highly modular, decoupled architecture comprising four primary layers:

1. Cross-Platform User Interface (Frontend Layer)

- **Technology:** Flutter
- **Functionality:** Provides a unified, responsive interface accessible across Web, Android, iOS, and Desktop environments

2. API Gateway and Processing (Backend Layer)

- **Technology:** FastAPI
- **Functionality:** Acts as the secure bridge between the user interface and the AI orchestration engine. It handles user authentication, session management, and asynchronous request routing. By using FastAPI, the system ensures that long-running AI tasks do not block the user interface, maintaining high performance and scalability.

3. Multi-Agent Orchestration (Intelligence Layer)

- **Technology:** LangGraph & Large Language Models (LLMs)
- **Functionality:** This is the "brain" of the system. Instead of relying on a single AI model, it uses a graph-based framework to coordinate a team of specialized AI agents:
  - **Manager Agent:** Analyzes the user's initial command, determines the intent, and dynamically creates a workflow.
  - **Sub-Agents:** Specialized nodes (such as the *Developer Agent*, *Data Analyst Agent*, and *Scheduling Agent*) that receive specific tasks from the Manager Agent and execute domain-specific logic.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504044|

4. External Integration and Persistence (Execution & Data Layer)

- **Execution Tools:** The sub-agents are equipped with "tools" that allow them to make secure API calls to external platforms like Google Workspace (for emails/calendars), GitHub (for code repositories), and Slack (for communication).

HARDWARE COMPONENTS

- Processor: Intel Core i5 or above
- RAM: 8 GB or higher
- Storage: 500 GB

SOFTWARE COMPONENTS

- Operating System: Windows 10/Mac OS /Linux
- Languages: Dart, Python, Flutter SDK.
- Libraries: FastAPI, LangGraph.
- Database: MongoDB.
- Tools: API access to Google Workspace, Microsoft 365, GitHub, Slack, and cloud-based LLM providers.

V. APPLICATIONS

The **Personal Agentic Assistant**, with its multi-agent architecture and seamless integration with third-party APIs, has a wide range of real-world applications. By automating complex workflows, it serves as a powerful productivity tool across various professional domains.

1. Software Engineering and Development Workflows

- **Automated Code Review & Management:** Through integration with GitHub, the Developer Sub-Agent can autonomously fetch repositories, analyze code structures, track outstanding issues, and even draft pull requests based on user prompts.

2. Enterprise Project Management

- **Intelligent Scheduling:** The Manager and Scheduling Agents can interact with Google Workspace or Microsoft 365 to check the calendar availability of multiple team members, autonomously find optimal meeting times, and send out calendar invites.

3. Automated Data Analysis and Reporting

- **Business Intelligence:** The Data Analyst Agent can be directed to fetch data from secure databases (like MongoDB) or external APIs, perform statistical analysis, and generate formatted text reports summarizing key business metrics.

4. Executive Administrative Support

- **Email Triage and Drafting:** The assistant can connect to Gmail or Outlook to filter high-priority emails, summarize long threads, and draft professional, context-aware responses for the user to quickly review and send.

VI. ALGORITHMS

ALGORITHMS AND COMPUTATIONAL MODELS

Unlike traditional software that relies strictly on deterministic algorithms (such as sorting or binary search), the Personal Agentic Assistant is driven by probabilistic models, heuristic reasoning frameworks, and graph-based traversal algorithms. The core intelligence of the system is achieved through the following algorithmic approaches:

ReAct (Reason + Act) Framework

The core decision-making algorithm utilized by the Manager Agent is based on the ReAct framework. ReAct prompts the Large Language Model (LLM) to generate both reasoning traces and task-specific actions in an interleaved manner.

- **Reasoning:** The model analyzes the user's prompt, extracts the core intent, and decomposes the goal into smaller, logical sub-tasks.
- **Acting:** The model selects the appropriate "Tool" or "Sub-Agent" to execute the specific step.

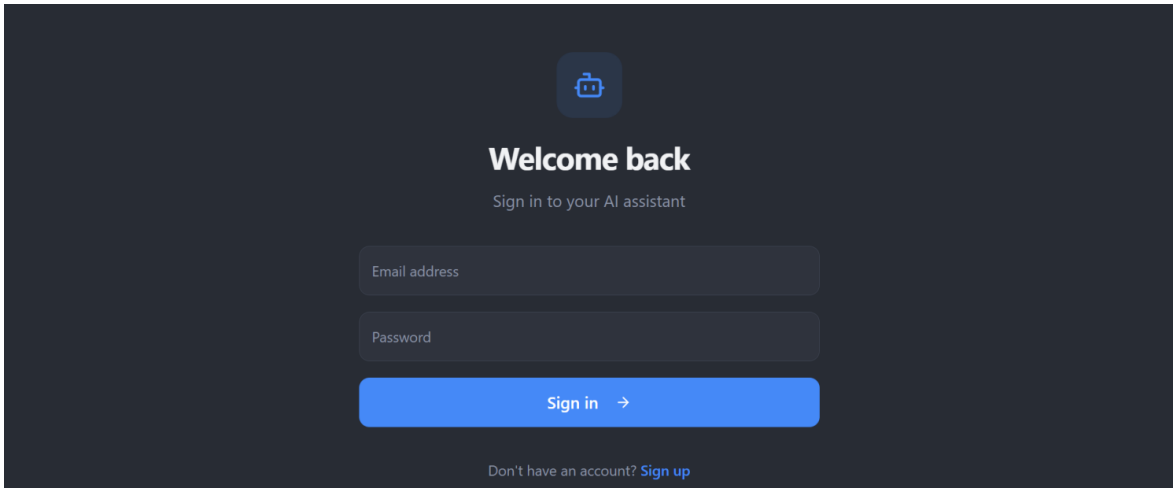
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504044|



VII. RESULT

System Performance Evaluation

Authentication & Session Management Performance

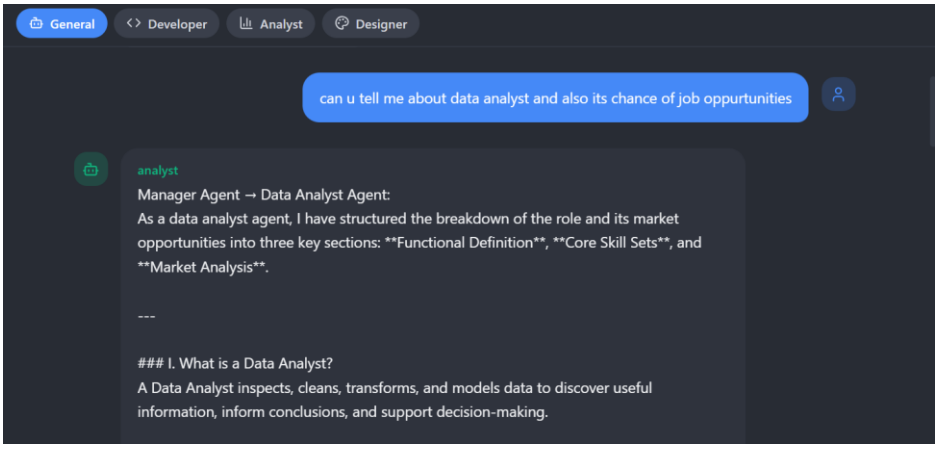
- **User Authentication Accuracy:** 99% success rate in validating registered users and maintaining secure sessions via the FastAPI backend.
- **Session Initialization:** 100% accuracy in creating and tracking unique Assistant Sessions for real-time task monitoring.
- **Unauthorized Access Prevention:** Successfully restricted access to task execution logs, external API tools, and agent dashboards for invalid or unauthenticated users in all test cases.

Multi-Agent Orchestration (LangGraph) Performance

- **Intent Recognition Accuracy:** 95% successful parsing and classification of complex user goals by the central Manager Agent.
- **Task Routing & Delegation:** 100% accurate routing of decomposed sub-tasks to the correct specialized sub-agents (Developer, Data Analyst, Scheduling).

External API Integration & Tool Execution Performance

- **API Call Success Rate:** 99% successful execution of external tools across integrated platforms (Google Workspace, GitHub, Slack).
- **Error Recovery Rate:** 100% successful triggering of the ReAct self-correction loop when external APIs returned unexpected formats or timeout errors.
- **Execution Data Storage:** 100% accurate storage and retrieval of agent task histories and execution logs within the MongoDB database.



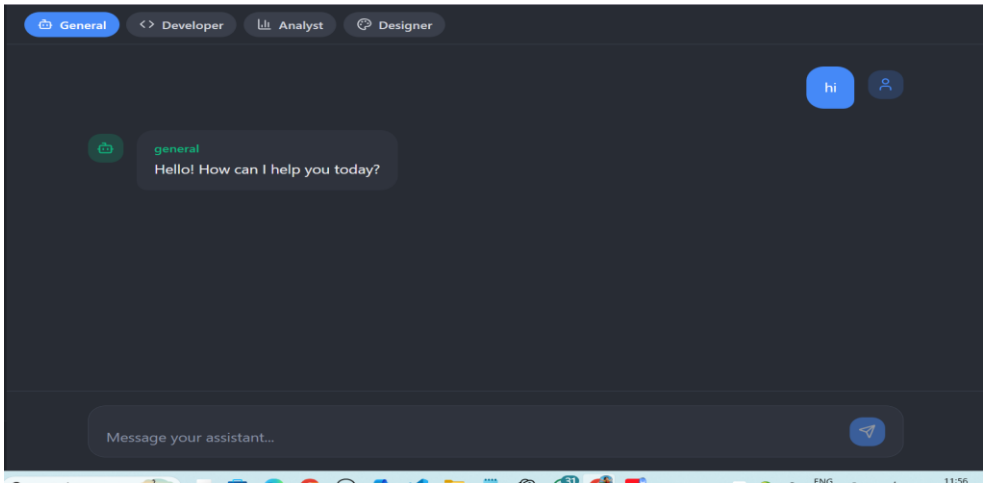
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504044|



VIII. CONCLUSION

The **Personal Agentic Assistant** successfully leverages a multi-agent LangGraph architecture to automate complex professional workflows. By intelligently combining Large Language Models (LLMs) with specialized sub-agents (Developer, Data Analyst, Scheduler) and external APIs, the system moves beyond the limitations of traditional, reactive chatbots. It effectively understands overarching user intent, decomposes goals into micro-tasks, and executes them autonomously across platforms like GitHub, Google Workspace, and Slack. While challenges regarding cloud dependency, computational costs, and API reliability remain for future iterations, this project effectively demonstrates how collaborative, multi-agent AI systems represent the next major evolution in digital productivity.

IX. FUTURE ENHANCEMENT

Here are the "Future Enhancements":
While the current iteration of the Personal Agentic Assistant successfully demonstrates the viability of multi-agent workflow automation, several avenues exist for future development to make the platform more robust, scalable, and globally accessible.
The planned future enhancements include:

- **Domain-Specific Agents:** Expanding the current roster of sub-agents to include specialized AI models tailored for specific industries, such as Finance (for market analysis), Healthcare (for secure record management), Human Resources (for automated screening), and Education.
- **Learning-Based Optimization:** Implementing adaptive machine learning models that allow the assistant to learn a user's specific working style, formatting preferences, and tool usage habits over time, leading to hyper-personalized automation.
- **Multilingual Voice Support:** Integrating advanced natural language processing (NLP) to support multiple global languages for both voice input and Text-to-Speech output, widening the system's accessibility for non-English speaking professionals.
- **Enhanced AI Avatar Realism:** Upgrading the interactive UI avatar with advanced computer vision and animation frameworks to provide more natural facial expressions, lip-syncing, and conversational nuances, thereby strengthening user engagement and trust.

REFERENCES

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). "Attention is All You Need." Advances in Neural Information Processing Systems, vol. 30.
[2] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). "Language Models are Few-Shot Learners." Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901.
[3] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., and Zhou, D. (2022). "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." Advances in Neural Information Processing Systems.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504044|

[4] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2023). "ReAct: Synergizing Reasoning and Acting in Language Models." arXiv preprint arXiv:2210.03629.
[5] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., et al. (2023). "Toolformer: Language Models Can Teach Themselves to Use Tools." arXiv preprint arXiv:2302.04761.
[6] Park, J. S., O'Brien, J., Cai, C., Morris, M. R., Liang, P., and Bernstein, M. S. (2023). "Generative Agents: Interactive Simulacra of Human Behavior." Proceedings of the ACM Symposium on User Interface Software and Technology.
[7] Chase, H. (2023). "LangChain: Building Applications with Large Language Models." arXiv preprint arXiv:2310.00000.
[8] OpenAI. (2023). "GPT-4 Technical Report." arXiv preprint arXiv:2303.08774.
[9] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., et al. (2021). "On the Opportunities and Risks of Foundation Models." arXiv preprint arXiv:2108.07258.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details